

---

Subject: Re: Is there a less clunky syntax for retrieving data from an array of hashes?

Posted by [David Fanning](#) on Thu, 29 Aug 2013 12:41:28 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

kagoldberg@lbl.gov writes:

>  
> In my opinion, that syntax clunkiness is a small price to pay for what you get by using hashes and lists. I program a lot with objects, and with objects, you have to declare and manage all of the fields explicitly. Every time I add a field, I have to put it in the `__define`, then in the `::Cleanup`, then in the `::GetProperty`, then in the `::SetProperty`, then in `::Init`, then write code to handle it properly. We all probably do these tedious mechanics. And unless you use pointers (talk about clunky), those object fields have a fixed type in advance.  
>  
> Now with hashes, you can do just about anything with any type of data, changing it with the freedom of a pointer, but without the cleanup hassles. Need a new hash field? Go ahead and just start using it on the fly. Does a certain field already exist? Just ask `.hasKey()`. Lists can `.Count()` `.Add` and `.Remove` in ways that are cleaner than the `IDL_Container()`'s way of doing things. And, you can very easily pass them around without copying memory, so every routine can share and modify the same hash flexibly, and changes appear in all places at once, if you catch my drift.  
>  
> Once you start using lists and hashes, you start programming in 'foreach' instead of 'for', and you save the preliminary steps of counting your elements before defining the loop range.  
>  
> And don't forget, hashes do have an elegant way to reach inside, as follows.  
> `h = hash()`  
> `h['a'] = [10,20,30]`  
> `print, h['a', 1]`  
> `h['a'] = findgen(5,10)`  
> `print, h['a', 2, 3]`  
>  
> Make sure to condition your data so you don't ask for index values out of range, and these tricks work very nicely.  
>  
> Thank you, IDL, for hashes and lists.

Thanks for this post.

I have to admit that lists and hashes tend to confuse me, and I have had a hard time figuring out what I would use them for. (Not to mention that using them would suddenly put the Coyote Library out of the reach of hundreds of users.) So, I read this article with a great deal of interest.

I think I understand most of your points, and I know as well as you the tedium of adding new fields to objects. But, as I was reading I got a queasy feeling in my stomach. Doesn't all this ad hoc adding of fields

and changing things around on the fly result in programs that are nearly impossible to understand and/or maintain? How do you manage that aspect of using lists and hashes?

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>

Sepore ma de ni thue. ("Perhaps thou speakest truth.")

---